Project Report

Data Storage Paradigms, IV1351

Nariman Haidar

narimanh@kth.se

2022-11-09

1 Introduction

The goal of this project is to create a database that will allow a music school to store all the necessary data for its operations. There are four sub-tasks that are related to this project. The first task is to create a conceptual model of the database. The second task is to a physical model. The third one is to build a program. The fourth task is to develop part of Soundgood's web site. I worked on this task with Aya Babiker.

2 Literature Study

Literature study that I used in this task was: The valuable resources teacher provided us on the canvas, the lectures by the teacher, book Fundamentals of database systems 7th edition, Elmasri and Navathe and YouTube.

3 Method

Before the group began working on the first task, each group member first studied the lectures, read the pertinent chapters of the book (ELMASRI, 2016), and watched the video on how to develop a domain model, IE notation, and conceptual model. We employed a program called Astah.

Then, in order to build a conceptual representation of the database, my partner and I simply begin by reading all the information. Because the goal of the first assignment was to create diagrams using all the data and the Entity-Relationship (ER). In order to create the database for "the Sound Good music school," we started by gathering all (noun identifying) Entities that would be relevant. Make a list of all entities that called category list search. We carried out each stage as shown in the conceptual model video.

Then my partner and I discussed every Entity we included in our model, deciding which ones should be deleted and which ones would be relevant to it. We provide the data types for the attributes, their cardinality, whether they can be empty or have a value, and whether they must be unique or not null. After that, we came up with the essential attributes and built the proper relationships between them.

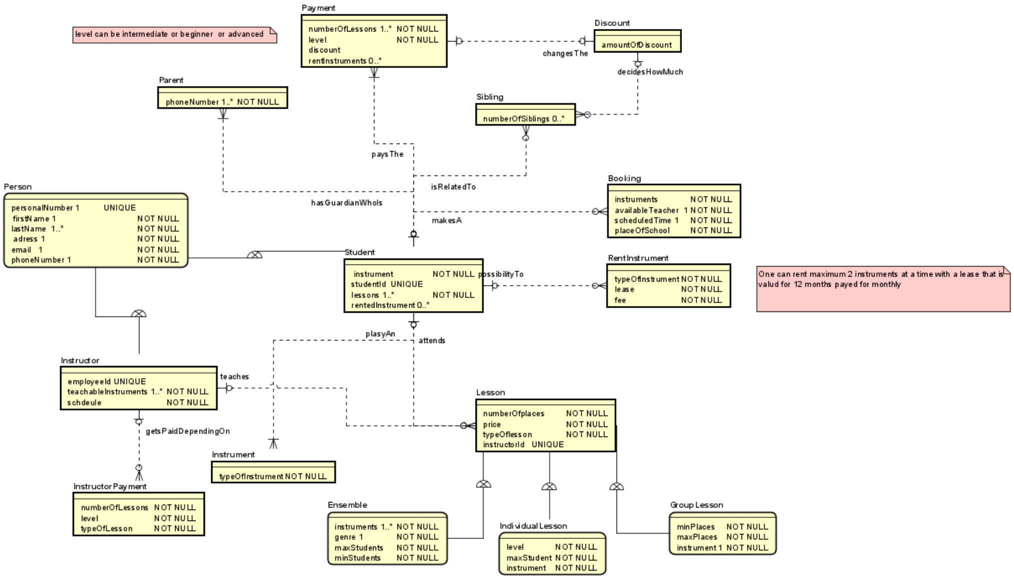
4 Result

The result is shown in Flowing Figure 1, which also includes the conceptual model and all the necessary information.

The person entity from whom the student and teacher are inherited is depicted in the model. Also, the lesson entity is inherited entity.

We have two distinct entities for payment according to how many lessons a student takes, whether or not they have rented an instrument from the school, and whether or not they have siblings enrolled in the school, a student's tuition may be discounted. And one for the instructor, who receives compensation from the institution based on the number of classes they provide.

We have a booking entity that manages all information regarding a school's location, available classrooms, schedule times, and instruments. We have an entity to lesson and all kinds of lessons. We have an entity to an instrument and all other entities just like shown in our result.



level can be intermediate or beginner or advanced

One can rent maximum 2 instruments at a time with a lease that is valud for 12 months payed for monthly

Figure 1: Conceptual model.

5 Discussion

We adhered to all guidelines provided in the canvas, such as naming conventions, avoided any mistakes, followed the instructions in Ramez Elmasri's "Fundamentals of Database Systems" to create a diagram without making any mistakes, and completed the IE diagram's steps. All the visible entities were represented in the diagram and we don't have any missing entities and every entity has attributes and attributes for all data. All attributes marked with not null or unique. we don't have any irrelevant relations Our diagram is very simple to grasp and follow and easy.

There are models with entities that do not inherit, and models with entities that do.

We decided to utilize the one with inheritance for ourselves.

The purpose of inheritance is to prevent the repeating of the same attributes across many entities. Therefore, I made use of the person entity, which has all the necessary information about a person, including name, ID, address, and other attributes so an attribute that is required for all lesson types only needs to be added once to an entity that is inherited. And the entity lesson is also an inheritance entity.

But in the first model without inheritance can function just as well, but it is ineffective because we must add the same information to each of the tree entities, which makes it difficult to maintain a database application.